

Linkmodel: gevit and cgevit

This is work in progress.

Parametrization

Hyperparameters

Specification

Hyperparameter spesification and default values

doc GEVIT link

hyper

theta1

hyperid 49033

name gev tail

short.name tail

initial 0.1

fixed FALSE

prior pc.egptail

param 5 -0.5 0.5

to.theta function(x, interval = c(REPLACE.ME.low, REPLACE.ME.high)) log(-(interval[1]

from.theta function(x, interval = c(REPLACE.ME.low, REPLACE.ME.high)) interval[1] + (

theta2

hyperid 49034

name gev p0

short.name p0

initial 0

fixed FALSE

prior normal

param 0 1

to.theta function(x) log(x / (1 - x))

from.theta function(x) 1 / (1 + exp(-x))

pdf gevit

doc Complement GEVIT link

hyper

theta1

hyperid 49035

name gev tail

short.name tail

initial -3

fixed FALSE

prior pc.gevtail

param 7 0 0.5

```

to.theta function(x, interval = c(REPLACE.ME.low, REPLACE.ME.high)) log(-(interval[1]
from.theta function(x, interval = c(REPLACE.ME.low, REPLACE.ME.high)) interval[1] + (
theta2
  hyperid 49036
  name gev p0
  short.name p0
  initial 0
  fixed FALSE
  prior normal
  param 0 1
  to.theta function(x) log(x / (1 - x))
  from.theta function(x) 1 / (1 + exp(-x))

```

pdf gevit

Example

```

n <- 3000
x <- rnorm(n, sd = .5)
intercept <- runif(1)
beta.x <- runif(1, 0.5, 1.5)
eta <- intercept + beta.x * x

xi <- -0.3
p.intercept <- inla.link.invgevit(intercept, tail = xi)
prob <- inla.link.invgevit(eta, tail = xi)
size <- 2
y <- rbinom(n, size = size, prob = prob)

r <- inla(y ~ 1 + x,
  data = data.frame(y, x),
  family = "binomial",
  Ntrials = size,
  control.inla = list(cmin = 0, int.strategy = "eb"),
  control.fixed = list(remove.names = "(Intercept)"),
  control.family = list(
    control.link =
      list(model = "gevit",
        hyper = list(tail = list(prior = "pcegptail",
          param = c(7, -0.5, 0.5)),
          intercept = list(initial = 0, param = c(0, 1))))),
  verbose = !TRUE)

summary(r)

round(dig = 3,
  cbind(true = c(p.intercept = p.intercept, beta.x = beta.x, xi = xi),
    estimate = c(p.intercept = r$summary.hyperpar[2,"mean"],
      beta.x = r$summary.fixed["x", "mean"],

```

```

xi = r$summary.hyperpar[1, "mean"])))

## this shows that the intercept is not part of the linear predictor, then also, not the fitted
## values
plot(eta, r$summary.linear.predictor$mean +
      inla.link.gevit(r$summary.hyperpar[2,"mean"],
                     r$summary.hyperpar[1,"mean"]),
      lwd = 3, col = "red", type = "l")
abline(a = 0, b = 1, lwd = 1, col = "blue")

##### same check for 'cgevit' link
##p.intercept <- 1 - inla.link.invgevit(intercept, tail = xi)
##prob <- 1 - inla.link.invgevit(eta, tail = xi)
p.intercept <- inla.link.invcgevit(intercept, tail = xi)
prob <- inla.link.invcgevit(eta, tail = xi)
## to get the same data
y <- size - y

rc <- inla(y ~ -1 + x,
           data = data.frame(y, x),
           family = "binomial",
           Ntrials = size,
           control.inla = list(cmin = 0, int.strategy = "eb"),
           control.family = list(
             control.link =
               list(model = "cgevit",
                    hyper = list(tail = list(prior = "pcegptail",
                                              param = c(7, -0.5, 0.5)),
                                intercept = list(initial = 0, param = c(0, 1)))))
print(round(dig = 3,
           cbind(true = c(p.intercept = p.intercept, beta.x = beta.x, xi = xi),
                       estimate = c(p.intercept = rc$summary.hyperpar[2,"mean"],
                                   beta.x = rc$summary.fixed["x", "mean"],
                                   xi = rc$summary.hyperpar[1, "mean"])))

```